STUK

STUK-YTO-TR 202 / MAY 2004

QUANTITATIVE RELIABILITY ASSESSMENT IN THE SAFETY CASE OF COMPUTER-BASED AUTOMATION SYSTEMS

Haapanen Pentti, Helminen Atte, Pulkkinen Urho
VTT Industrial Systems

In STUK this study was supervised by Marja-Leena Järvinen

STUK • **SÄTEILYTURVAKESKUS**
STRÅLSÄKERHETSCENTRALEN
RADIATION AND NUCLEAR SAFETY AUTHORITY

Osoite/Address • Laippatie 4, 00880 Helsinki
Postiosoite / Postal address • PL / P.O.Box 14, FIN-00881 Helsinki, FINLAND
Puh./Tel. (09) 759 881, +358 9 759 881 • Fax (09) 759 88 500, +358 9 759 88 500 • www.stuk.fi

The conclusions presented in the STUK report series are those of the authors and do not necessarily represent the official position of STUK

ISBN 951-712-835-5 (print)
ISBN 951-712-836-3 (pdf)
ISSN 0785-9325

Dark Oy, Vantaa/Finland 2004

HAAPANEN Pentti, HELMINEN Atte, PULKKINEN Urho (VTT Industrial Systems). Quantitative reliability assessment in the safety case of computer-based automation systems. STUK-YTO-TR 202. Helsinki 2004. 36 pp.

Keywords: safety case, reliability assessment, automation, computer-based system, programmable systems, protection systems, nuclear reactor safety, Bayesian networks

Abstract

An essential issue in the construction of new or in the replacement of the old analogue automation applications in nuclear power plants is the reliability of computer-based systems, and especially the question of how to assess their reliability. The reliability issue is particularly important when the system under assessment is considered as a safety-critical system, such as the reactor protection system. To build sufficient confidence on the reliability of computer-based systems appropriate reliability assessment methods should be developed and applied. The assessment methods should provide useful and plausible reliability estimates, while taking the special characteristics of the reliability assessment of computer-based systems into consideration.

The Bayesian inference has proved to be an efficient methodology in the reliability assessment of computer-based automation applications. Practical implementation of Bayesian inference, Bayesian networks, allow the combination of the different safety arguments concerning the system and its development process to a unified reliability estimate. Bayesian networks are also a convenient way to communicate on the safety argumentation between various participants of systems design and implementation as well as between the participants in the licensing processes of computer-based automation systems.

This study is a part of the research project “Programmable Automation System Safety Integrity assessment (PASSI)”, belonging to the Finnish Nuclear Safety Research Programme (FINNUS, 1999–2002). The project aimed to provide support for the authorities and utilities in the licensing problems of computer-based automation systems. Particular objective of the project was to acquire, develop and test new and more cost-effective methods and tools for the safety and reliability assessment, and to gather practical experience on their use in order to achieve a more streamlined licensing process for the computer-based automation systems.

The project is financed together by the Radiation and Nuclear Safety Authority (STUK), the Ministry of Trade and Industry (KTM) and the Technical Research Centre of Finland (VTT).

HAAPANEN Pentti, HELMINEN Atte, PULKKINEN Urho (VTT Tuotteet ja tuotanto). Ohjelmoitavien automaatiojärjestelmien kvantitatiivinen luotettavuusarviointi järjestelmien lupakäsittelyssä. STUK-YTO-TR 202. Helsinki 2004. 36 s.

Avainsanat: turvallisuusanalyysi, luotettavuusanalyysi, automaatio, ohjelmoitavat järjestelmät, tietokonepohjaiset järjestelmät, suojausjärjestelmät, reaktoriturvallisuus, Bayes-verkot

Tiivistelmä

Keskeinen kysymys ydinvoimalaitosten uusien ohjelmoitavien järjestelmien rakentamisessa tai vanhojen analogisten järjestelmien uusinnoissa on näiden järjestelmien luotettavuuden arviointi. Erityisen tärkeä tämä kysymys on turvallisuuskriittisten, esim. reaktorin suojausjärjestelmien osalta. Riittävän uskottavuuden saavuttamiseen tähtääviä luotettavuusarviointimenetelmiä tulee näin ollen kehittää ja soveltaa. Arviointimenetelmien tulee tuottaa käyttökelpoisia ja uskottavia luotettavuusarvioita ottaen huomioon ohjelmoitavien järjestelmien erityispiirteet.

Bayesilainen päättely on osoittautunut tehokkaaksi menetelmäksi ohjelmoitavien automaatiojärjestelmien luotettavuuden arvioinnissa. Bayesilaiseen päättelyyn perustuvien Bayes-verkkojen avulla voidaan yhdistää erilainen järjestelmää ja sen kehitysprosessia koskeva turvallisuusargumentaatio yhtenäiseksi luotettavuusarvioksi. Menetelmä toimii myös tehokkaana argumentoinnin välineenä ohjelmoitavien järjestelmien suunnittelun, toteuttamisen ja lisensiointiprosessien eri osapuolien välillä.

Tutkimus on osa Suomen kansalliseen ydinturvallisuustutkimusohjelmaan (FINNUS 1999–2002) kuuluvaa ”Ohjelmoitavan automaation turvallisuuden arviointi (PASSI)”-projektia. Projektin yleisenä tavoitteena on ollut tukea viranomaista ja voimayhtiöitä ohjelmoitavien automaatiojärjestelmien lisensointitehtävissä. Erityisesti on pyritty hankkimaan, kehittämään ja kokeilemaan uusia kustannustehokkaita menetelmiä ja työkaluja turvallisuuden ja luotettavuuden arviointiin, sekä keräämään käytännön kokemuksia menetelmien soveltamisesta ohjelmoitavien järjestelmien lisensiointiprosessin kehittämistä varten.

Projektia ovat rahoittaneet yhdessä Säteilyturvakeskus (STUK), Kauppa- ja teollisuusministeriö (KTM) ja Valtion teknillinen tutkimuskeskus (VTT).

Contents

1 Introduction

The safe operation of a nuclear power plant requires that the probability of accidents leading to a large release of radioactivity is extremely low [1]. That requirement means that the physical barriers (fuel, cladding, primary pressure system and the containment) preventing the release are kept intact with a high probability. Present Finnish nuclear power plant Guide YVL 2.8 sets quantitative probabilistic safety goals for a severe core damage and for a large release of radioactivity [2]. In present day non-passive plant concepts the safety requires certain active safety functions that can be started and carried through when needed with high reliability.

The use of software in different functions has increased rapidly and nowadays computer-based systems have become an integral part of nearly every engineering application. The additional functionality provided by software has been implemented also to the modern automation systems and these systems have successfully been used for example for the controlling of conventional thermal power plants. Good experiences on the use of computer-based automation systems and, on the other hand, the technical and economical ageing of the current automation systems in the existing nuclear power plants is in favour of replacing the systems with corresponding computer-based systems also in nuclear power plants.

For the demonstration of the fulfilment of the above overall safety goals an estimate of the reliability of the essential safety-critical automation applications is needed. This estimate cannot be based entirely on deterministic criteria since our present day level of knowledge would then lead to over-conservative assumptions and unbalanced design solutions. Probabilistic and quantitative criteria must therefore be included in the reliability estimation of computer-based automation systems.

One of the essential issues when replacing the automation applications of nuclear power plants is the reliability of computer-based systems, and especially the question of how to assess the reliability. The reliability issue is particularly important when the system under assessment is considered as a safety-critical system such as the reactor protection system. To build sufficient confidence on the reliability of computer-based systems appropriate reliability assessment methods should be developed and applied. The assessment methods should provide useful and plausible reliability estimates, while taking the special characteristics of the reliability assessment of computer-based systems into consideration.

A problem causing extra work in the reliability assessment of computer-based safety-critical systems has at least the following characteristics. First, strict reliability goals are set for the automation systems responsible for the safety functions of nuclear power plants and to demonstrate the achievement of these goals the systems should be well built and thoroughly tested. Second, the discontinuous behaviour of discrete logic in computer-based systems has the effect that to be sure on the functionality of the system in all occasions the system should be tested with all possible inputs. However, full testing of a system is usually not feasible because of the large number of possible inputs even for a relatively simple system and, therefore, for a more complicated computer-based system a thorough testing would require an unacceptable amount of time and effort. One proposal to overcome the problem is to compensate the shortage of testing with reliability related evidence from other sources closely involved with computer-based systems.

Research on reliability of computer-based systems has been carried out for several years in Finnish national research programs on nuclear

safety. In the Programmable automation systems in nuclear power plants (OHA) -project (1995–1998) e.g. the statistical testing methods and operational profiles and diversity requirements were studied and a reference model for safety case processes was developed for computer-based systems. Acquisition, development and testing of new and more cost-effective reliability and safety assessment methods for the computer-based systems have been the main objectives of the Programmable automation system safety integrity assessment (PASSI) -project belonging to the Finnish research programme on nuclear power plant safety (FINNUS) -programme (1999–2002). The emphasis of PASSI-project has been on the application of Bayesian inference in reliability assessment and an experimental case study was performed in the project. Description of the whole research carried out in PASSI-project is given in the final report of FINNUS-programme [3].

This report first describes the basic features of the safety case for computer-based safety critical automation systems. Following an illustration of the reliability modelling of computer-based systems using Bayesian networks is given. Further on the experimental application case study is summarised. In the end of the report a discussion on the gained experience and further research and development needs is given.

2 Safety case of computer-based systems

A safety related system must have a safety case; this is explicitly required by licensing regulations in a wide range of industries and equivalent requirements are given in many standards such as IEC 61508 [4]. The safety case should:

- demonstrate an adequate level of safety
- ensure safety is maintained throughout the lifetime of the system
- minimise project risk

Adelard [5] defines a safety case as:

A document body of evidence that provides a demonstrable and valid argument that a system is adequately safe for a given application and environment over its lifetime.

In the following some important aspects of the safety case are shortly discussed.

2.1 Acceptance criteria

Two complementary safety analyses have generally been used in the safety evaluation of nuclear installations. The different analyses are named as deterministic and probabilistic safety analysis. The deterministic approach is fully based on rules and guides established by rulemaking or regulatory organisations. In a report by De Gelder [6] the deterministic approach is illustrated with following major steps:

- identification and categorisation of events considered in the design basis of the installation under analysis;
- analysis of enveloping scenarios;
- evaluation of the consequences verification that acceptance criteria are met.

In probabilistic approach the analysis is taken beyond the design basis of the installation, and possible accident scenarios are evaluated with the extensive use of probability calculus. The major steps in the probabilistic safety analysis are following [6]:

- identification of the initiating events and the plant operational states to be considered;
- analysis of the possible accident scenarios, by means of event trees;
- reliability analysis, by means of fault trees, of the system considered in the fault trees;
- collection of probabilistic data (failure probabilities, unavailability for test and maintenance, initiating event frequencies)
- human error analysis;
- accident sequence quantification, resulting in a frequency for each core melt sequence;
- interpretation of the results (including sensitivity and importance analysis)

After all preventive and mitigating measures considered for the design of an installation using the deterministic safety analysis, the installation still represent some residual risk. This risk is considered in the probabilistic safety analysis.

In the case of computer-based automation systems the residual risk is particularly problematic. As mentioned already in the introduction to be sure on the functionality of the system in all occasions the system should be tested with all possible inputs in all internal system states. In practice this is usually not feasible because of the large number of possible inputs and system states even for a relatively simple system.

Given the special characteristics of computer-based systems it is reasonable to consider three types of acceptance criteria for the safety critical computer-based automation systems. The criteria can broadly be classified in three principal categories:

- deterministic,
- qualitative, and
- probabilistic.

A criterion is *deterministic* if its fulfilment can be expressed as a two-valued (i.e. true–false) logical expression. Even if it in practice may sometimes be difficult, this evaluation can always be made. Of the basic safety design principles the redundancy is a typical example of a deterministic criterion; a system/function realisation either includes redundancy or does not. The same goes also for the defence–in–depth safety principle. Separation/segregation is usually also considered to be a deterministic criterion, but in practice these also include some probabilistic features (e.g. is the separation of redundancies into different fire departments efficient against all plausible fire situations). Many single requirements of standards and guidelines are of deterministic nature, e.g. the requirement that interrupts shall not be used in the software of a safety critical programmable system.

A criterion is *qualitative* if the fulfilment can only be somehow graded on a qualitative scale. The adherence to standards and guidelines in general is of this type. Although the fulfilment of many of the single requirements, as seen before, can be evaluated exactly there still remains many that can only be assessed on a qualitative scale (e.g. very low, low, medium, high, very high) and usually requires some kind of expert judgement.

A criterion is *probabilistic* if the fulfilment can only be expressed as a probability or probability distribution. Of the basic safety design principles the diversity may at first sight seem to be a deterministic criteria; the system/function either includes diverse features or does not. As a safety design principle the diversity, however, does not have any absolute value, but introducing diverse features in a system is aimed for diverse failure behaviour of the system. There is some evidence, especially for computer-based systems that diverse redundant channels can fail simultaneously and only probabilistic assessment of their common cause failure propensity is possible.

For the clearness and unambiguity of the safety case a well-defined set of deterministic acceptance criteria would be desirable. Concerning the computer-based safety systems our present level of knowledge about the relation between the safety of the system and the deterministic features of the system and its design process is still somewhat inadequate. Strictly deterministic criteria would, therefore easily lead to unbalanced design solutions and wasteful resource allocation in the design and licensing of these systems. Qualitative and probabilistic acceptance criteria are therefore needed.

On the other hand, one should also bear in mind that the safety itself is a probabilistic concept. Safety is defined by nature as the freedom from unacceptable risk and risk is defined as the combination of the probability of occurrence of harm and the severity of the harm. The newly reformed Finnish safety regulation does not set any more direct reliability requirements on single safety functions, but only for the frequency of severe reactor accidents and large radioactive release. Plant design solutions then lead to corresponding reliability requirements for those plant functions credited in plant probabilistic safety analyses.

2.2 Safety evidence

The licensing process is defined as a set of interrelated activities, whose objective is to collect evidence supporting the safety claims about the system to be licensed, and based on this evidence to assess, ideally numerically, the achieved reliability or safety of the system. Figure 1 gives an example of a safety demonstration process for a computer-based safety automation system in United Kingdom. [7]

The quality evidence about the design process consists e.g. of adherence to well-established set of standards and guidelines, use of qualified design, verification and validation tools, skilled personnel, high quality project management and quality control/quality assurance program, documentation

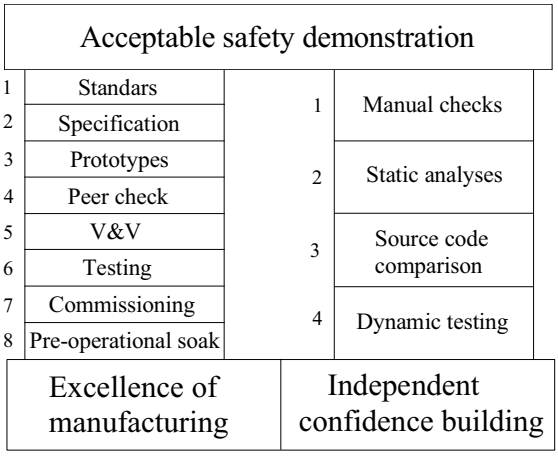

Figure 1. Safety demonstration process (UK).

etc. This evidence shall be raised in parallel with the proceeding the design process itself. It will evolve and become more detailed and accurate when the system design shapes up. This mostly qualitative evidence is then combined during the licensing process with the high quality product evidence gained through tests, analyses and operational experience.

Presently the safety systems are in most cases implemented on a pre-qualified, or certified, automation system platform. In this case the design process is divided in two completely separate phases, namely:

- platform design process (providing the platform hw/sw-components and tools for the application design and V&V activities)
- application design process (providing the application)

The acceptance evidence can accordingly be divided to process and product evidences concerning the platform and the application as shown in Figure 2.

The platform approach can provide advantages both in costs and safety:

- more effort can be put in the platform design process when development costs can be divided between a large amount of applications; this will increase the process and product quality,
- operating experience from different applications, if properly collected and used for product improvements, can improve the product quality,
- platform evidence is the same for each individual application making a type acceptance process possible and reducing the costs of the application licensing process,
- individual application may, however, require some adjustments for the platform components or tools, and make project specific evaluation of these necessary, and
- platform evidence, of course, can not alone guarantee the safety of the application, but other pieces of evidence presented in fig. 2 are needed as well.

A central issue in the licensing process is the combination of the different pieces of evidence mentioned above to a safety and reliability estimate of the application. Actually inference from statistical

	PLATFORM	APPLICATION

PRODUCT	**What evidence we have about the platform?**
	• *Artifacts from the lifecycle*
	  • *Requirements, different kind of analysis, design documentation, source codes, test reports...*
	• *3rd. party qualification*
	• *Certificates, requirements and results*
	• *Version histories and operational experiences*
	• *etc...*
		What evidence we have about the application?
		• *Artifacts from the lifecycle*
		  • *Requirements, different kind of analysis, design documentation, source codes, test reports...*
		• *etc...*

PROCESS	**What evidence we have about the platform development process?**
	• *Quality documents of the platform development process*
	  • *V&V-reports, 3rd. party assessments etc..*
	• *Resources*
	  • *Personnel, education...*
	• *Configuration mgmt*
	• *Tools used and tool quality*
	• *etc...*
		What evidence we have about the application development process?
		• *Quality documents of the application development process*
		  • *V&V-reports, 3rd. party assessments etc..*
		• *Resources*
		  • *Personnel, education...*
		• *Configuration mgmt*
		• *Tools used and tool quality*
		• *etc...*

Figure 2. Safety evidences for a platform application safety system.

testing is the only sound method available for estimating software reliability. However, if one ignores evidence other than testing, e.g., evidence from the track record of the developer, or from the quality of the development process, the results are going to be so conservative that they are often felt to be useless for decision-making. Bayesian inference is the main mathematical tool for taking into account diverse knowledge. Closer introduction to Bayesian inference and Bayesian networks is given later in the text.

2.3 Failures and defences

Understanding the failure mechanisms in computer-based systems and the knowledge about the defences against them is a necessity for the definition of the structure and contents of the safety case. Figure 3 describes the basic failure mechanisms in such a system.

Malfunction is a result of some fault in the system. Faults are either inherent in the system, if they can be attributed to deficiencies in the production of the system, e.g. to errors in system design or to flaws arising in the implementation/ manufacturing process, or they can be caused by random hardware component failures. The faults lead either to immediate malfunction or they can stay latent in the system until a specific triggering input leads to a conditional malfunction.

Hardware component failures may occur either due to ageing (wear and tear, physical or chemical effects) or due to some harmful external influences.

The influence of failures due to ageing on the reliability of the system can be determined with adequate certainty by means of usual reliability engineering methods. Due to the extensive operational experience available, the underlying fault models and data are of high quality. Redundancy, i.e. the provision of multiple parallel channels, is the basic mean to protect the system functioning against random component failures.

Failures due to external influences can occur in all equipment exposed to such influences. Experience has shown the following to be the most common external influences:

- climatic conditions (heat, moisture),
- mechanical impacts (e.g. earthquake),
- heat and/or smoke from fires, and
- electromagnetic fields.

The approach of creating physically separated building sections or equipment trains ensures that the maximum scope of impact of any such failure is restricted to one building section/equipment train. Due to the structural design and arrangement of the plant buildings faults for example in the nuclear steam supply system cannot affect the central automation equipment. Further precautions (e.g. selective fusing) may be taken to confine the reflection of a failure to smaller, defined areas within a building section or equipment train (e.g. in a single electronic equipment cabinet). Protection against external impacts such as flooding, lightning strikes and storms must be provided not just in the form of local separation but also by means of further structural design features for all items of safety automation equipment (e.g. linking the cabinets by optic fibres).

Malfunctions due to production deficiencies are caused by faults that are inherent in the system. Only duly qualified, individually shop-tested com-

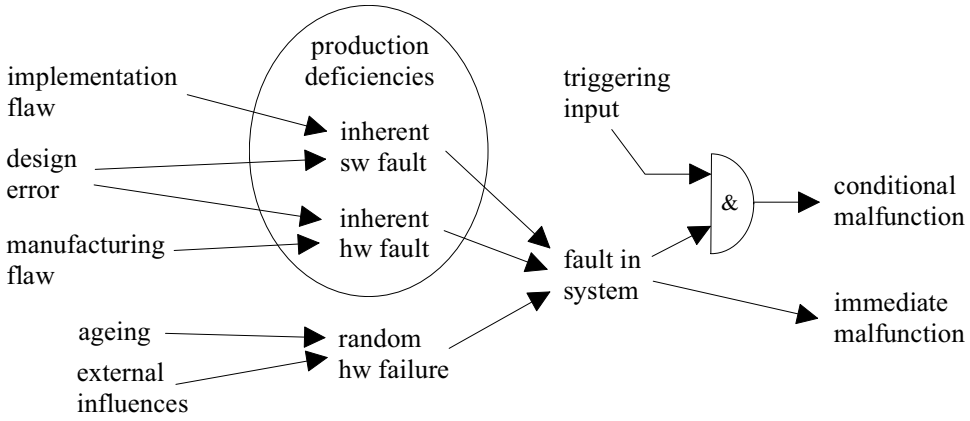

Figure 3. Failure mechanisms in a computer-based system.

ponents should be used in safety automation systems. Shop and system tests are performed to verify that the equipment actually exhibits the specified quality and design features. In particular, the system test verifies that equipment fulfils its design requirements. But even the most intensive testing cannot completely rule out the inherent faults.

2.3.1 Hardware and software faults

It is a characteristic of faults due to production deficiencies that they are present in the system at all times. This implies that, given identical initial conditions and triggering inputs, faults due to production deficiencies will always cause malfunctions in the same way. As presented in Figure 3 the product deficiencies are caused by implementation flaws, design errors and manufacturing flaws in hardware and software. In the following, a distinction between the characteristics of hardware and software faults will be made.

Computer–based automation systems are made up of only a few different types of hardware modules that are used together with different software for specific applications. The hardware is used in time–sharing mode by numerous functions implemented in the software, which access the hardware in identical cycles. The engineering effort that goes into the hardware, and consequently the potential for hardware faults, are considerably lower than in the case of hardwired automation systems. In particular, cyclic accessing of the hardware makes it highly probable that faults due to production deficiencies in computer–based automation systems will cause spontaneous malfunction or – if the faulty feature is not used – will not cause any malfunction at all. It should be pointed out that experience to date provides no evidence that hardware faults due to production deficiencies in the instrumentation and control equipment could cause this equipment to fail as a result of circumstances arising in the power plant process. Faults are particularly improbable in the light of the fact that strictly cyclic processing of all programs used during normal operation, which is a basic design requirement, and ensures that the status of the instrumentation and control system is independent of any challenges made to the system. That is to say, there are no known mechanisms that could lead to such a linked response.

The reason for system failures, caused by software, cannot be either in ageing or in wear and tear of the software. Software can change only if its properties are intentionally changed. The reason of false behaviour is already in software when it is being installed and software faults are introduced into programs during the design process, either by the designers or the tools used. Inherent software fault is activated to cause the system malfunction by some input signal combinations with certain internal system states. Taking into account that the internal state of the software is a deterministic function of the input signal history, or input trajectory, this actually means that the unreliability of the software is determined by the probability of the occurrence of the error activating signal trajectory i.e. the operation profile of the system. Two identical computer-based systems can have quite different reliabilities if they are exposed to different operation profiles. This fact e.g. makes it harder to utilise operational experience in the reliability assessment of computer-based systems.

2.3.2 Types of software faults

Considering the development process of programmable automation systems, the importance of early phases of the process is even more emphasised as in ordinary software development. Here the software implementation oriented phases are not so problematic, as applications are automatically built of quality controlled software components. Of course, errors may also be created during application building phases, but usually the tools are well validated, via extensive use for instance, and this possibility is often less likely than human errors in earlier phases. In any case, the following types of software faults in programmable automation system can be distinguished:

- faults due to errors in the requirement specification,
- faults due to errors in the functional specification,
- faults due to errors in code generation.

Software faults due to errors in the requirement specification applies to specified requirements which are either inappropriate for the identification of the incidents and accidents concerned or which provide no suitable countermeasures for

control of these. Such errors cannot be avoided by technical means, but a formalised design process, and in particular effective verification and validation methods make it much more probable that such errors will be discovered. A software engineering process, which applies interpretable specification documents, thus supporting validation with the aid of a proven plant simulator makes for a considerable improvement in quality by comparison with design engineering for hardwired safety systems.

A technique, which is also of major importance in counteracting the effects of errors in the requirements specification derived from the process design, is diversity. The means to implement diversity on the one hand in the form of diverse criteria for identification of an accident malfunction with associated diverse countermeasures, and on the other hand in a defence–in–depth approach. Suitably structured automation systems and appropriate distribution of functions to different equipment can help to ensure that diversity in the system requirements is also implemented in the form of diverse software and thus constitutes an effective means of accommodating software faults originating in other phases of development.

Software faults due to errors in the functional specification are the result of incorrect translation of the system requirements into the automation systems functional specification. Besides the need to accommodate the effects of such faults, the use of quality–assured measures in the translation of the system requirements into the functional specification in order to avoid such faults in the first place is of particular importance.

Software faults due to errors in code generation are the result of incorrect translation of the formal functional specification into the object code for the digital computer. In principle, such errors can have very different effects. The measures that can be taken to avoid and to accommodate them are just as varied as the effects of such errors. The key to avoiding such failures lies in the method of automatic code generation and in the capacities and quality of the tools used. For this reason, code generation for computer–based automation systems should always be performed with the aid of tried and tested tools. The quality of these tools will be enhanced as the user base expands, thanks to the produced improvement resulting from large–scale experience feedback.

2.3.3 Common cause failures

Software faults are design faults introduced to the software during the design and implementation process. In a multi–channel redundant system having identical software in each channel also the possible residual faults are present in all channels. If an input sequence (signal trajectory) triggering erroneous response is fed to all channels they may fail simultaneously. The residual software faults can thus be a source of common cause failures (CCF's) of a redundant programmable system.

There is also some evidence that even channels having diverse software may contain common design faults that cause them to fail simultaneously. Software CCF potential may exist between different systems or between different channels in one system e.g. when common software modules are used. Other common features with CCF potential include common architecture, algorithms, development methods, tools, implementation methods, staffing and management [8].

Deficiencies in software can be due to incorrect, incomplete, inaccurate or misunderstood software requirements and software specifications. Design errors or software faults can be present in diverse programs, due to common human factors such as training, organisation, thinking processes and design approaches.

In summary, the experiments conducted on this issue indicate that statistically correlated failures result from the nature of the application, from similarities in the difficulties experienced by individual programmers, and from special cases in the input space. The correlations seem to be related to the fact that the programmers are all working on the same problem and that humans do not make mistakes in a random fashion.

There is no reason to expect that the use of different development tools or methods, or any other simple technique, will reduce significantly the incidence of errors giving rise to correlated failures in multiple-version software components. All evidence points to the fact that independently developed software that uses different programmers, programming languages, and algorithms but computes the same function (satisfies the same functional requirements) cannot be assumed to fail in an independent fashion.

Also abnormal hardware failures, plant condi-

Table I. Failure types and defences.

Fault types	**Random component faults**	**Inherent design faults**	**External events**
Failure types	**Single failure**	**Can lead to Common Cause Failures (CCF)**	
Defence	Redundancy	Diversity	Separation/segregation
	1oo2 sufficient for safety, but can cause inadvertent actuations. These lower the plant availability and can also act as initiating events for accident scenarios 2oo3 sufficient both for safety and availability but reduces to 1oo2 when one train is failed/ under maintenance. 2oo4 manages all the previous if no common cause failures present	Proper selection of hardware, software and implementation diversity can protect the redundancies against common cause failures caused by inherent faults. Functional diversity is considered a good defence against common cause failures in diverse redundancies.	Physical separation and geographical segregation protects the redundancies to fail due external events. Functional separation prevents the spreading of failure effects from the failed redundancy to others.
Total defence	Diversity and separation/segregation between redundant trains		
Residual problem	Common cause failures in diverse redundancies? Solution: Functional diversity (= Defence in Depth)?		

tions and events can cause unforeseen software states, transients or overload conditions that were not covered by the initial requirements or by the software design. Abnormal or failed states of hardware and plant can cause software failure in a channel or functional path. Where redundant channels are used, an abnormal or failed state, which appears on two or more channels, can cause software CCF. Where two systems are used whose functions are diverse, the abnormal or failed state can appear in both functional paths and cause software CCF.

A more comprehensive discussion about the application of different diversity principles in programmable safety systems can be found e.g. in [9].

2.3.4 Defences for different failure types

In conclusion one can state that if it is not possible to achieve the desired reliability concerning inherent software faults in a single channel system (the containment of hardware failures may still require multiple redundant channels) some kind of diversity is necessary. Independently developed (from common functional specifications) computer-based channels can improve the system reliability, but it is hard to assess the achieved improvement. A better assurance can be achieved either by using functionally diverse computer-based channels or having an analog back-up system for the computer-based channel(s). Use of analog back-ups, however, divests much of the benefits of the transition to computer-based system technology and should be avoided as far as possible. Defences for different failure types are summarised in Table I.

The common cause failure (CCF) propensity (see Chapter 2.3.3) of the computer-based automation system applications is a central issue in designing and implementing of safety critical automation functions of nuclear power plants. The diversity principle is a commonly accepted design strategy for solving this problem; the application of different forms of diversity and the evaluation of the achieved reliability, however, still poses many open questions.

On the higher level the Finnish regulatory system requires that:

"In ensuring the most important safety functions, systems based on diverse principles of operation shall be used to the extent possible (395/1991, Guide YVL 1.0)."

In present day, non-passive plant designs this means that even the automation functions participating in the realisation of these functions shall have diversity; regardless of the selected technology (electric, electronic or programmable electronic).

On a single safety function level it is the designers task to apply the above mentioned redundancy, diversity and separation principles in

such a way that – among other things – the reliability requirements set for the function (see Chapter 2.4) are fulfilled. This is a typical design optimisation task where design, implementation and licensing costs are balanced in order to achieve an acceptable safety level at minimal costs.

In principle the designer has the following three levels of diversity and various combinations of these available in his design task; hardware, software and functional diversity:

These can further be divided in sublevels:

- Hardware diversity
  - different computers or processors
  - different automation platforms
  - analogue back-up systems
- Software diversity
  - separate design teams
    - random diversity
    - enforced diversity
  - different design and implementation tools (compilers, linkers, loaders, operating systems etc.)
- Functional diversity
  - division of the function into diverse subfunctions with the same overall safety goal

The influence of these various diversity alternatives was shortly discussed in Chapter 2.3.3. In conclusion one can say that there so far is rather limited experience on the effectiveness of various diversity approaches and further research is still needed.

2.4 Function and system requirements

The safe operation of a nuclear power plant requires that the probability of accidents leading to a large release of radioactivity is extremely low [1]. That requirement means that the physical barriers (fuel, cladding, primary pressure system and the containment) preventing the release are kept intact with a high probability. Present Finnish Nuclear power plant Guide YVL 2.8 sets goals for a severe core accident (less than 10^{-5} /year) and for a large release of radioactivity (less than $5 \cdot 10^{-7}$ / year) frequencies [2].

In present day – not entirely passive – plant concepts the protection of the integrity of release barriers requires certain highly dependable active safety functions that can be started and carried through when needed with high reliability. The dependability of a safety system is composed of the following factors shown in Figure 4.

- Firstly, the automatic safety system shall include proper functions needed for the detection of anomalous plant process states and for the starting of countermeasures for returning the process back to a safe state and for mitigation of the consequences of accident situations.
- Secondly, these functions shall have the required performance, that is, the ability to realise the necessary functions correctly and effectively. The performance can further be divided in factors of capacity, fault tolerance and reliability.
- Finally, the reliability is factored to the environmental durability of the system hardware and absence of design faults both in hardware and software.

The task of the process design and deterministic process safety analyses (transient and accident analyses) is to define what automatic functions are needed for the prevention of radioactive releases during normal plant operation, anticipated transients and postulated accident situations. Deterministic analyses also define the necessary timing, capacity etc. performance requirements for these functions.

Probabilistic process safety analyses, on the other hand, define the necessary reliability requirements for the safety functions in order to fulfil the quantitative safety goals given in Guide YVL 2.8.

These deterministic and probabilistic requirements for safety functions establish the input for the automation engineering, whose task is to allocate the requirement to systems and equipment so

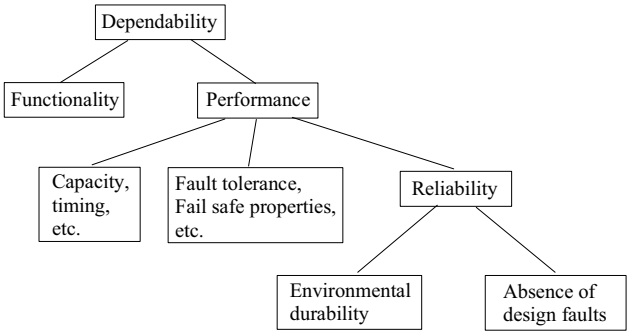

Figure 4. Factors of the system dependability.

that all functional and performance (including reliability) requirements will be satisfied with high confidence. The progression of different design tasks is illustrated in Figure 5.

The essence of the safety case of a computer-based safety critical automation function and associated system and equipment (FSE) is to demonstrate with the satisfactorily high confidence that the object of the case fulfils all requirements set on it, including the fault tolerance, fail safe etc. requirements. It is obvious, therefore, that complete, clear and unambiguous requirement specifications are a necessity for the safety case.

2.5 SHIP-model of the safety case

The basic objective of the licensing process is to collect evidence supporting the safety claims about the automation system to be licensed, and based on this evidence to assess, ideally numerically, the achieved reliability or safety of the system. As systems become more complex, this becomes increasingly difficult. Complexity increases the risks of both random component failures and design-related failures. Incorporating redundancy in the design can mitigate random hardware failures. Design-related failures, such as software faults, cannot be mitigated in the same way, as the design fault would be common to redundant components, so design faults may become the dominant factor affecting the safety of a complex system.

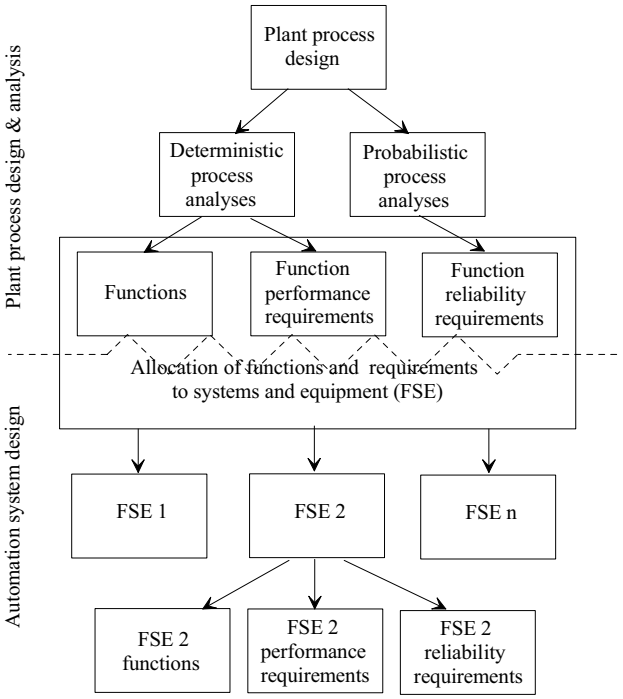

Figure 5. The progression of design tasks for safety automation systems.

For some most safety critical automation systems in nuclear power plants quantitative reliability targets are set by the licensing authority. For random hardware failures there are well-established techniques for quantifying their reliability implications. The assessment of the impact of design faults is more difficult. The main problem with this quantification is that we do not know, in advance, the number and nature of the design faults remaining in the system so it is difficult to quantify their impact on reliability.

The estimation of the reliability of programmable automation systems, possibly containing (software) design faults, is then largely based on evidence on quality of the design process. High quality design process:

- minimises the introduction of faults in the system,
- detects and removes the faults during the design process, and
- uses proper means of tolerating the residual faults.

This mostly qualitative evidence is then combined during the licensing process with other evidence about the system gained through tests, analyses and operational experience.

The quality evidence about the design process shall be raised in parallel with the proceeding the process itself. It will evolve and become more detailed and accurate when the system design shapes up. At each stage, the basis of the safety arguments should be clear and as stated by Bishop et. al. [10] they should:

- make an explicit set of claims about the system,
- provide a systematic structure for marshalling the evidence,
- provide a set of safety arguments that link the claims to the evidence,
- make clear the assumptions and judgements underlying the arguments and
- provide for different viewpoints and levels of detail.

Using a model drafted in the assessment of the Safety of Hazardous Industrial Processes in the presence of design faults (SHIP) -project (financed

by EEC in the framework of the Environment programme, sub-theme: Major Industrial Hazards) a safety case consists of the following elements [11]:

- claims about properties of the system or subsystem
- evidence which is used as the basis of the safety argument
- arguments linking the evidence to the claims
- inference rules that provide the logical basis for the steps in an argument

The structure of the SHIP-model is summarised in Figure 6. The evidence in the figure could in fact be a sub-claim so the whole argument structure is recursive, hiding the details in lower level arguments. The evidence might also be initial design assumptions, which have to be supported by confirmatory tests as the development proceeds.

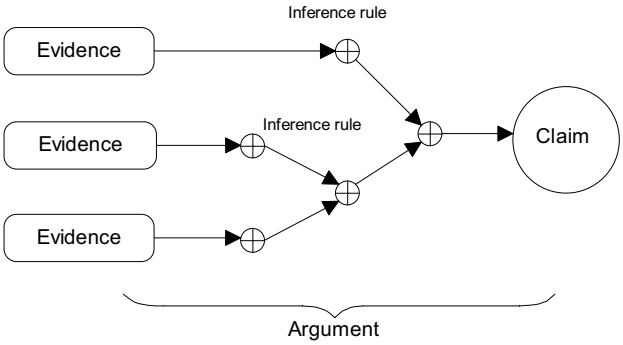

Figure 6. SHIP-model for the safety argumentation.

The actual nature of the argument and the inference mechanism can vary depending on the system design and the licensing process strategy. For example, an argument could be:

- Deterministic, where the evidence can be axioms, the inference mechanism is the rules of predicate logic, and the safety argument is a proof using those rules.
- Probabilistic, where the evidence could be component failure rates and assumptions of independence, and the inference mechanism is statistical analysis.
- Qualitative, where the evidence might be adherence to standards, design rules, or guidance. The inference mechanism is some form of acceptance criterion based on this.

In addition, the overall argument should be robust, i.e. the argument should be sound even if there are uncertainties or errors in parts of the argument. Also, it is evident that expert judgement will have a significant role in the safety argumentation. To have a consistent way of implementing the expert judgements to the safety argumentation suitable assessment methods should be developed. In the following chapters a method developed for taking an advantage of expert judgements to utilise the SHIP-model is described.

3 Reliability modelling using Bayesian networks

The practical reliability assessment method developed and applied in PASSI-project aims to support the conceptual approach of SHIP-model presented in the previous chapter. The general research approach has been to model and combine reliability evidence of computer-based automation systems using the framework of Bayesian modelling, and in particular the technical solution of Bayesian modelling called Bayesian networks.

The Bayesian modelling framework was chosen as a basis of reliability assessment for different reasons. First of all, Bayesian networks provide a formal and consistent approach for modelling and combining different kinds of evidence in a same reliability analysis where the uncertainties in the evidence can be expressed using the mathematical framework of probability calculus. Second, in a Bayesian model the reliability of a system is explicitly stated so that basis of the reliability estimation can easily and transparently be verified.

The principles of Bayesian modelling in the reliability assessment of computer-based systems are presented in a reliability modelling process described below. The reliability modelling process was tested in a case study on a quantitative reliability estimation of a software-based motor protection relay. The emphasis of the case study was on the methodological analysis of the assessment approach and summary of the case study is given in the following chapter. Before going to the reliability modelling and assessment of computer-based systems in detail a general introduction to the world of Bayesian modelling and Bayesian networks is given.

3.1 Bayesian statistics and interpretation of probability

Reliability is usually measured in terms of probability and, in general, statistical methods are used to estimate these probabilities. Probability is an abstract notion formulated as a measure theoretic model for uncertainty. In a pure mathematical formulation probability is an additive measure defined in a measure space. The mathematical formulation doesn't say anything about the interpretation of probability but only fixes the rules of probability calculus. Different statistical frameworks give different interpretation of probability and these interpretations should be carefully considered when making inferences based on probabilities of certain statistical framework.

The statistical framework most often exploited in the probability estimation is based on the frequentistic interpretation of probability, and the frequentistic interpretation of probability is the one most often introduced to students in the elementary courses of statistics. The frequentistic interpretation aims to an objective view of probability, which is defined as a long run frequency of certain phenomena. Through the frequency definition probabilities in the frequentistic interpretation are considered as objective properties of natural phenomena, which do not depend on the observer. The probabilities in the frequentistic interpretation are estimated using the methods of classical statistics.

Another interpretation of probability is the subjective interpretation. In the subjective interpretation of probability the probabilities are interpreted as degrees of beliefs on the occurrence of certain events or truths of certain proposals. The statistical approach adopted in the subjective interpretation is the Bayesian statistics. In the Bayesian statistics all unknown parameters of a model are considered as random parameters and the uncertainties of the unknown parameters are expressed with prior probability distributions. The structure and the parameters of the model are modelled by their joint distribution and the statis-

tical inference on the parameters of interest is carried out following the likelihood principle, i.e. all information contained by a sample is described by the likelihood function. Often Bayesian statistics is described as predictive way of carrying out statistical inference since the statistical inference is applied on the observed information to predict the future. More detailed introduction to the theory and use of Bayesian statistics in reliability estimation of computer-based systems is given for example in reports by Korhonen et al. [12] and Helminen [13].

3.2 Bayesian modelling and Bayesian (belief) networks

Bayesian modelling is a consistent formal modelling framework based on Bayesian statistics. With Bayesian modelling the system can be divided unambiguously into subparts or phases, which can later be combined using the elegant theory of Bayesian statistics. The term Bayesian in the modelling framework is used to emphasise two particular characteristics. First, the term Bayesian refers to a consistent use of Bayesian statistics and to the subjective interpretation of probability as described above. Second, the objective of the Bayesian modelling framework is to build a joint probability distribution for the system under study. The model provides a comprehensive view of the whole system under modelling not only a small subpart. Of course, the modelling can be focused on a smaller subpart or even a single parameter of the system. The joint probability distribution is then used to estimate the properties of the subpart or the parameter.

Bayesian networks provide a graphical tool for Bayesian modelling. Intuitively a Bayesian network is a description of a Bayesian model, where a collection of nodes is connected with directed arcs forming a network. The network is a graphical representation of the joint distribution of a model, where the nodes represent the parameters and the arcs describe the dependencies between the parameters. Bayesian networks provide an easy and flexible method for the modelling of complex systems involving inexact information. The efficient way to illustrate and form joint probability distributions for a large set of variables along with the increasing number of available Bayesian network computer programs have increased the interest toward Bayesian modelling. In the report the application of Bayesian networks to the reliability estimation of computer-based systems is discussed. The Bayesian approach has also been widely applied in other modelling applications and to review for example the use of Bayesian networks in the research area of adaptive and intelligent systems see a report by Myllymäki and Tirri [14].

3.3 Bayesian networks in reliability modelling

Traditionally, the reliability estimation of a computer-based system has had strong reliance on the evidence acquired from the system testing. To demonstrate the achievement of strict reliability goals set for example for the safety-critical applications of nuclear power plants the systems should be well developed and built, and thoroughly tested. With computer-based systems the difficulty is that the number of possible tests even for a relatively simple system may be enormous and full testing of the computer-based system may not be feasible. For safety-critical systems plausible reliability estimations are, however, needed and one proposal to overcome the difficulty is to compensate the shortage of testing evidence with evidence from other sources. The evidence, or potential sources of evidence, in the reliability estimation of computer-based systems were shortly reviewed in Chapter 2.

While different evidence is used in the reliability estimation of computer-based systems the combination of evidence becomes an essential and important part of the assessment. Different evidence involves different characteristics, and to be able to combine the pieces of evidence together extra flexibility from the assessment methods is required. The flexibility requirement favours the use of Bayesian networks in the reliability estimation. Bayesian networks provide a flexible, transparent and consistent way of modelling and combining different kind of evidence in a single reliability assessment.

According to the SHIP-model presented in the previous chapter the licensing process of safety-critical systems can be seen as a process of setting up safety claims on properties of a system and then through analysing and combining evidence the claims are argued. In similar manner the

reliability estimation of computer-based systems using Bayesian networks is carried out. First, the collected evidence is analysed and formulated suitable for the Bayesian modelling framework. Second, combining the different pieces of evidence using Bayesian networks a statistical inference on the reliability of a computer-based system is concluded. The outcome of the process is a model providing not only a reliability estimate, but also an easy access to the evidence and to the beliefs building up the estimate.

In practice, the evidence analysis and evidence combination in the reliability modelling process cannot be separated and the two parts are carried out side by side. A general description of the reliability modelling process using Bayesian networks in the reliability estimation of computer-based systems is given below. In the description the main phases of the modelling process are illustrated and relevant inputs and outputs of the Bayesian network model are explained. The main phases of the modelling process are also present in the case study of the following chapter.

3.4 Reliability modelling process

The purpose of a reliability modelling process is to develop a concise model for the estimation of system reliability. Generally, the reliability model and the reliability estimate are used to support the decision-making on the applicability of the system for a certain purpose. The strong interrelationship between the reliability estimation and the decision-making has an influence to the reliability modelling process and to the reliability assessment in general.

The general decision-making process is formulated so that at the beginning of the process the decision situation is outlined by identifying the decision objectives and alternatives. Next, the decision problem is decomposed and modelled. In the model the problem structure, uncertainties and preferences are determined. Based on the model the best alternative for the decision problem is chosen and sensitivity analysis for the alternative is carried out. The final step is to evaluate whether further analysis is needed, and if not, then implement the chosen alternative.

Partly inspired by the general decision-making process the main phases of a structured modelling process applied in the project for the reliability estimation of computer-based systems are depicted in Figure 7. The modelling process is composed of three main phases: model structuring, model application and sensitivity analysis. Along with

PHASES	INPUTS	OUTPUTS
I. Reliability Model Building	– Definition of system parameters and data: $\boldsymbol{\theta}$, $\mathbf{x}$ – Definition of conditional probability distributions: $p(\theta_i\|\theta_1,\ldots,\theta_{i-1},\theta_{i+1},\ldots,\theta_n)$	– Likelihood function: $p(\mathbf{x}\|\boldsymbol{\theta})$
II. Reliability Model Application	– Definition of prior distributions: $p(\theta_i)$ – Collection of data: $\mathbf{x}$	– Joint distribution: $p(\boldsymbol{\theta},\mathbf{x})$ – Posterior distributions: $p(\theta_i\|\mathbf{x})$
III. Sensitivity Analysis of Model	– Variation of likelihood function: $p(\mathbf{x}\|\boldsymbol{\theta})$ – Variation of prior distributions: $p(\theta_i)$ – Variation of data: $\boldsymbol{x}$	– Analysed posterior distributions: $p(\theta_i\|\mathbf{x})$

Figure 7. Diagram representing the phases of reliability modelling process.

the main phases the different inputs and outputs of the Bayesian networks are presented in the diagram. In following, the different phases of the modelling process are discussed in more detail.

3.4.1 Model structuring

In statistical inference the fundamental idea is to use observed data to learn about the unknown features of a system. The unknown features are modelled in the form of random parameters. In order to make inference on the parameters it is essential to describe the links between the parameters and the data. Denoting the data by $\boldsymbol{x}$ and the parameters by $\boldsymbol{\theta}$, the objective is to use $\boldsymbol{x}$ to learn about $\boldsymbol{\theta}$. For example in the case study the failure rate of different software versions of the motor protection relay was the estimated parameter $\boldsymbol{q}$, and the amount of operation years and the number and types of software defects encountered for the different software versions were used as data to carry out the estimation.

The inference between data and parameters is usually carried out using a statistical model. The statistical model is defined by stating the joint distribution of data, model parameters and model structure. The model structure, which determines the links between data and model parameters, is defined in the model through conditional probabilities.

In Bayesian statistics, the joint distribution $p(\boldsymbol{x},\boldsymbol{\theta})$ is a product of a likelihood function $p(\boldsymbol{x}\,|\,\boldsymbol{\theta})$ and a prior distribution $p(\boldsymbol{\theta})$. The use of prior distribution in Bayesian inference is the main source of disagreement between the two schools of interpretation of probability described above. The prior distribution formulates person's prior belief about the parameters, and therefore the natural interpretation of probability in Bayesian statistics is the subjective interpretation. In the case study the prior distributions were formed for the failure rate of the first software version and for the reliability changes between successive software versions.

The likelihood function is presented as a conditional distribution of the observed variables given the structure of the model and the other parameters. Loosely speaking the likelihood function can be described as the probability to observe the data $\boldsymbol{x}$. In the case study the likelihood was constructed by using Poisson distribution for the number of failures occurred in certain time interval.

In practise, the reliability modelling process using Bayesian networks is usually carried out so that a group of conditional probability distributions $p(\theta_i \,|\, \theta_1,\ldots,\theta_{i-1},\theta_{i+1},\ldots,\theta_n)$ determining the interrelations between the parameters are defined. Combining the conditional probability distributions together the likelihood function is defined and the model structure is fixed. The knowledge for the model structuring is generally gathered from the reliability theory, from different characteristics of computer-based systems and their development process. Also, previous experiences and examples of similar reliability modelling processes can provide valuable help for the model structuring.

In the model-structuring phase it is important to analyse the computer-based system and its environment thoroughly and ground the claims proposed in the model to the true characteristics of the system using the valid rules of reliability theory. The result of the model-structuring phase is a Bayesian network setting a frame for the reliability estimation. In the network definitions for the parameters and for the dependencies between the parameters are explicitly stated.

3.4.2 Model application

In the application phase of the reliability modelling process the evidence on the reliability of computer-based system is introduced to the model. As described above, the model variables were divided to data and unknown parameters. In practice, the division is not so self-evident and it is more convenient to consider all variables of the model as unknown parameters. Actually, in Bayesian statistics all parameters are considered as random variables where the observed parameters become fixed. The parameters can therefore be grouped as observable parameters and so-called hidden or auxiliary parameters, which cannot be valued or observed directly. Evidence of the observable parameters is implemented to the model as data to carry out inference on the hidden parameters.

The hidden parameters may represent physical quantities that for some reason cannot be observed or they may be abstractions of the reality that can only be measured in conceptual level. Estimated failure rate in the case study is a good example of a hidden parameter of the second kind. Reliability of a system can be measured in different ways depending on the system and the circum-

stances. Failure rate, i.e. the number of failures per time unit, is often used. Failure rate cannot be measured in same sense as for example the physical quantities of temperature or pressure. However, estimates for the failure rate can be evaluated by making an inference based on the observable parameters of a system such as number of failures and time of operation of the system.

Due to the special characteristic of the hidden parameters their evidence is usually sparse or the interpretation and implementation of the evidence to the model is ambiguous. In Bayesian networks the hidden parameters are given prior estimations presenting subjective judgements on the values of the parameters. In the reliability estimation of computer-based systems the prior estimations are normally formed in some sort of expert judgement process using evidence from the development process and the design features of the system. The uncertainties about the values of the hidden parameters are modelled by probability distributions.

Building the prior distributions of the model parameters is an important part of the model application phase. Since the prior estimations are subjective beliefs on the values of the parameter before the observation of the data, it is inevitable that different people will have different beliefs and will come to different conclusions about the prior distributions. To avoid prejudice and arbitrariness it is important to make the assessment of the prior distributions as transparent as possible. The prior belief should be formulated so that they can be defended to other people as being based on genuine knowledge. In the case study the building of the prior reliability distribution was based on an expert judgement process described in the following chapter.

After the prior distributions for the hidden parameters are estimated, data for the computer-based system can be collected and implemented to the model. Data provides additional information about the hidden parameters, and the prior estimates are updated using the Bayes' rule. The updated distributions are called posterior distributions, and they are the conditional distributions of the hidden parameters given the evidence. The posterior distributions describe the uncertainty about the hidden parameters when the information from the observed parameters is taken into account.

The result of the model application phase is the joint distribution of the reliability model, from which the posterior distributions of the hidden parameters can be integrated. For example in the case study the posterior distributions under interest were the probability distributions of the failure rates of the different software versions produced in the life cycle of the motor protection relay. In relatively simple networks the calculations can be carried out using conventional numerical techniques, but for networks containing tens and hundreds of parameters the calculation must be carried out using Monte Carlo Markov Chain methods. Fortunately, nowadays there are many commercial and non-commercial computer programs available for the modelling and calculation of Bayesian networks.

3.4.3 Sensitivity analysis

The final phase of the reliability modelling is the sensitivity analysis phase. The purpose of sensitivity analysis is to determine how sensitive the reliability model is to the assumption made in the evidence and model building. Despite the importance of the sensitivity analysis the limited resources of the modelling process may sometimes cause a temptation to leave the phase for a small consideration only. This would be a major drawback since the purpose of the sensitivity analysis phase is not only to investigate how sensitive the results are to the assumptions but also to make judgements about the rationality of the results and the reliability model in general.

Typical subjects of sensitivity analysis in Bayesian networks are the prior distributions of the system parameters and the likelihood function, i.e. the dependencies between the model parameters. For prior distributions the sensitivity analysis is straightforward and can easily be carried out by calculating the posterior distributions using different prior distributions. For likelihood functions the analysis a bit harder since it usually requires changes in the structure of the Bayesian network. Sometimes it is necessary to carry out a sensitivity analysis to the data as well. For example in the case study the interpretation of data on the software defects wasn't unanimous, and therefore it was reasonable to carry out the calculations using two different data sets to see if the different interpretations had an influence to the final results.

As a result of the sensitivity analysis phase the posterior distributions and the reliability model are evaluated so that the uncertainties of the assessment are recognized and their influences to the results can be estimated. The consistent use of probability distributions in the reliability modelling process is a major benefit for the sensitivity analysis phase. Adjusting the variance of the probability distributions of the model parameters the uncertainty of the parameters can easily be modified. Careful execution of the sensitivity analysis phase gives support for the decision-making whether the evidence and reliability model is sufficient and no further analysis is needed.

3.5 Elicitation and summarisation

When making a statistical analysis of any kind, it is usually not the statistician who is directly interested in the results. Most often, the analysis is performed for a client who is somehow involved or interested in the results. In Bayesian modelling process the results are given in the posterior distributions. In fact, the posterior distributions contain all the information known about the hidden parameters θ at the moment. However, this is not always very helpful. The information is encapsulated in the mathematical formula, and meaningful summaries of the posterior distributions should be prepared. These summaries include means, modes or medians, variances and any other descriptive statistics that will help the client to see what is known about θ.

Just like summaries can be used as a natural way of expressing posterior information, summaries can also be used in the forming of prior distributions. In fact, the reliability modelling process presented above can be extended to a general reliability assessment process using Bayesian networks. The general assessment process is illustrated in Figure 8. The description of the reliability modelling process concludes the right hand side of the reliability assessment process. In the assessment process the reliability evidence is first collected to prior summaries. The prior summaries form the basis of elicitation to build up the prior distributions of the hidden parameters of a Bayesian model. After the prior distributions are determined, Bayes' rule is applied to update the prior distributions to posterior distributions. Finally, the relevant information of the posterior distributions is summarised to the client in the posterior summaries.

In the assessment process the left hand side represents the domain of the client, while the right hand side is the domain of the statistician. Summaries are the natural language to elicit the client's prior information to the statistician, and on the other hand to summarise the posterior information back to the client. The arrows in the figure represent the statistician's tasks in the assessment process. The case study in the following chapter is a practical example of a reliability assessment process described in Figure 8.

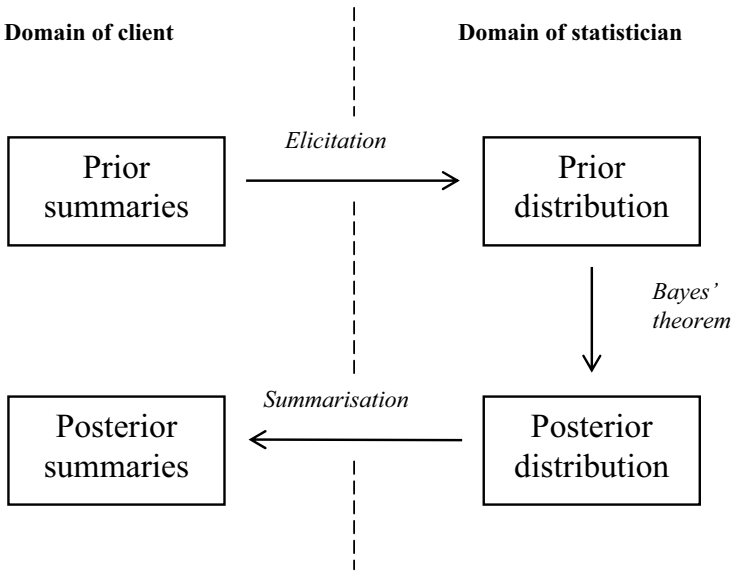

Figure 8. Representation of general reliability assessment process using Bayesian networks. Representation originates from the lecture notes of O'Hagan [15].

4 Case study on reliability estimation of a computer-based motor protection relay

To test the reliability assessment approach in practice a case study on a quantitative reliability estimation of a software-based motor protection relay was carried out in the project. The framework of Bayesian modelling and Bayesian networks was applied throughout the assessment. The emphasis of the assessment was on the methodological analysis of the assessment method, and the case study system and the evidence were reviewed only on a level necessary to test the functionality of the assessment method.

The parameter under estimation in the assessment was the failure rate of different software versions of the motor protection relay. Both quantitative and qualitative evidence were used to carry out an inference on the failure rate. The evidence was provided by the system manufacturer and prior summaries in the assessment were gathered in co-operation with the system developers using an expert judgement process described later on. Posterior summaries of the assessment are concluded in the graphs presenting in the end of the chapter. The presentation of the case study is a summary of a longer report and for more detailed description of the assessment and the numerical values used in the estimation see the report by Helminen & Pulkkinen [16].

The general structure of the summary recalls the main phases of the reliability modelling process introduced in Chapter 3. The summary starts with an explanation of the target system and the assessment process. Next, the structure of the reliability model is described. After the description of the model a detailed introduction to the evidence and to the application of the reliability model is given. Review on the sensitivity analysis and the results are given in the end of the chapter.

4.1 Target system and assessment process

The case study system under estimation was an integrated design of current measuring multifunctional relay for the complete protection of alternating current motors. The relay continuously measures the three phase currents and the residual current of the protected object. When a fault occurs and persists long enough to exceed a set or calculated occurrence time, the relay starts and operates. Depending on the relay setting and the fault the relay either gives an alarm or launches a protection signal to protect the object.

At the time of the assessment the case study system had not been applied in safety critical functions of nuclear power plants and explicit consideration if the relay fulfils all the aspects of a safety critical system of a nuclear power plant was not carried out in the assessment. However, the motor protection relay is a computer-based application for which high reliability is required, and therefore it was an excellent case study system for the assessment.

The idea in the assessment process was to combine the available evidence to form an as plausible reliability estimation of the target system as possible. In the assessment the whole life cycle of the software-based system was taken into consideration and a diagram representing the elements of assessment process is depicted in Figure 9.

In the beginning of the assessment a prior reliability estimation for the first software version of the relay was built. The prior estimation was based on the expert judgements on the product development process and it was constructed in an expert judgement process. The prior estimation

was then updated using the operational experience of the first software version. Later on, when the software was modified, the effects of the modifications to the software reliability were evaluated using expert judgement on the version management and the estimation was updated with the operational experience of the second software version. This procedure was repeated, as many times as there were new software versions produced in the life cycle of the system.

4.2 Structure of reliability model

The reliability model was mainly based on the characteristics of the available evidence in the assessment. The operational experience consisted of the approximated amount of working years and the amount and types of software defects encountered for the different software versions of the system. Therefore, the appropriate conditional distribution of the number of software faults of a software version given the failure rate and operating years was considered Poisson distributed as follows:

$$f(x|\lambda,T) \sim Poisson(\lambda T), \quad \text{(Eq. 1)}$$

where x is the number of software faults, l failure rate and T estimated total operating years of a software version. For more detailed introduction of Poisson processes in the reliability estimation of computer-based systems see for example book by Musa et al. [17].

The prior distribution for the failure rate of the first software version given by the experts was a mixture of lognormal distributions. In order to simplify the numerical computation in the assessment a log-transformed parameter of the failure rate was used, i.e. $\theta = ln(\lambda)$, where parameter θ is the log-transformed failure rate.

The failure rate of a software version in the model is constant, although unknown. However, the failure rate changes when shifting to the following software version. The change in failure rate is expressed in terms of log-transformed parameters as follows:

$$\theta_{i+1} = \theta_i + \omega_{i+1}; \quad \omega_i \sim N(\mu_i, \sigma_i^2) \quad \text{(Eq. 2)}$$

where θ_i and θ_{i+1} are the log-transformed failure rates of successive software versions and w is the random normal distributed change. Parameters μ_i and σ_i^2 correspond to the prior knowledge about the failure rate change, and were determined based on the expert judgement of the assessment executives.

4.3 Application of reliability model

The evidence in the assessment was based on two primary sources. The first source of evidence was the expert judgements of the system developers and assessment executives. The second source was the operational experience of the computer-based system. In the assessment the development process and version management of the system was reviewed by the experts and based on their judgements a prior failure rate distribution of the first software version and prior estimates of the reliability changes between the successive software versions were concluded. In the construction of the prior failure rate distribution a specific expert judgement process was carried out.

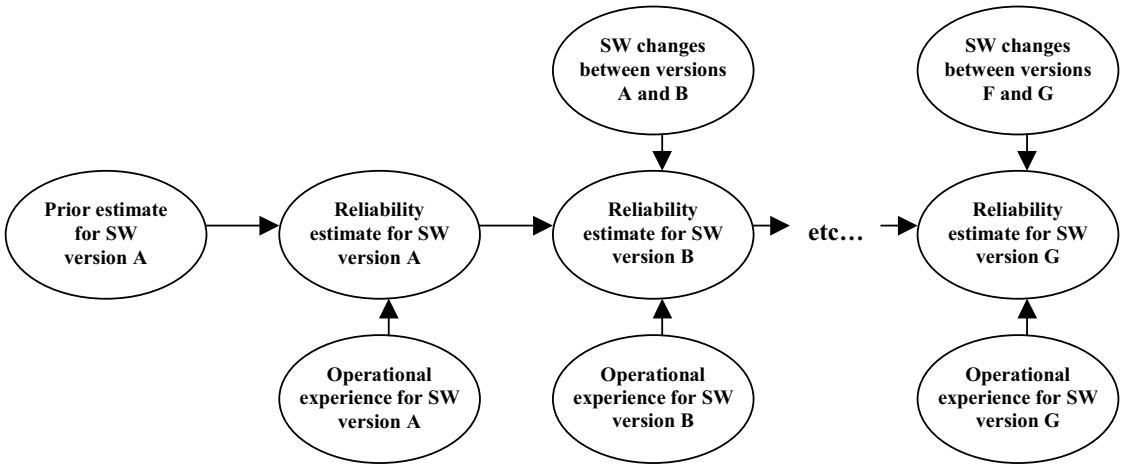

Figure 9. Diagram representing the software (SW) reliability assessment process.

4.3.1 Prior reliability estimate

The prior failure rate distribution of the first software version was built using the expert judgements of the product development personnel, such as project managers, designers and programmers of the system. The prior estimation was constructed in an expert judgement process. A diagram representing the six steps of the expert judgement process is illustrated in Figure 10.

In practice the expert judgement process was carried out using a collection of interviews. The purpose of the interviews was to recall the memories of the system development process. The questions of the interviews were divided to five catego-

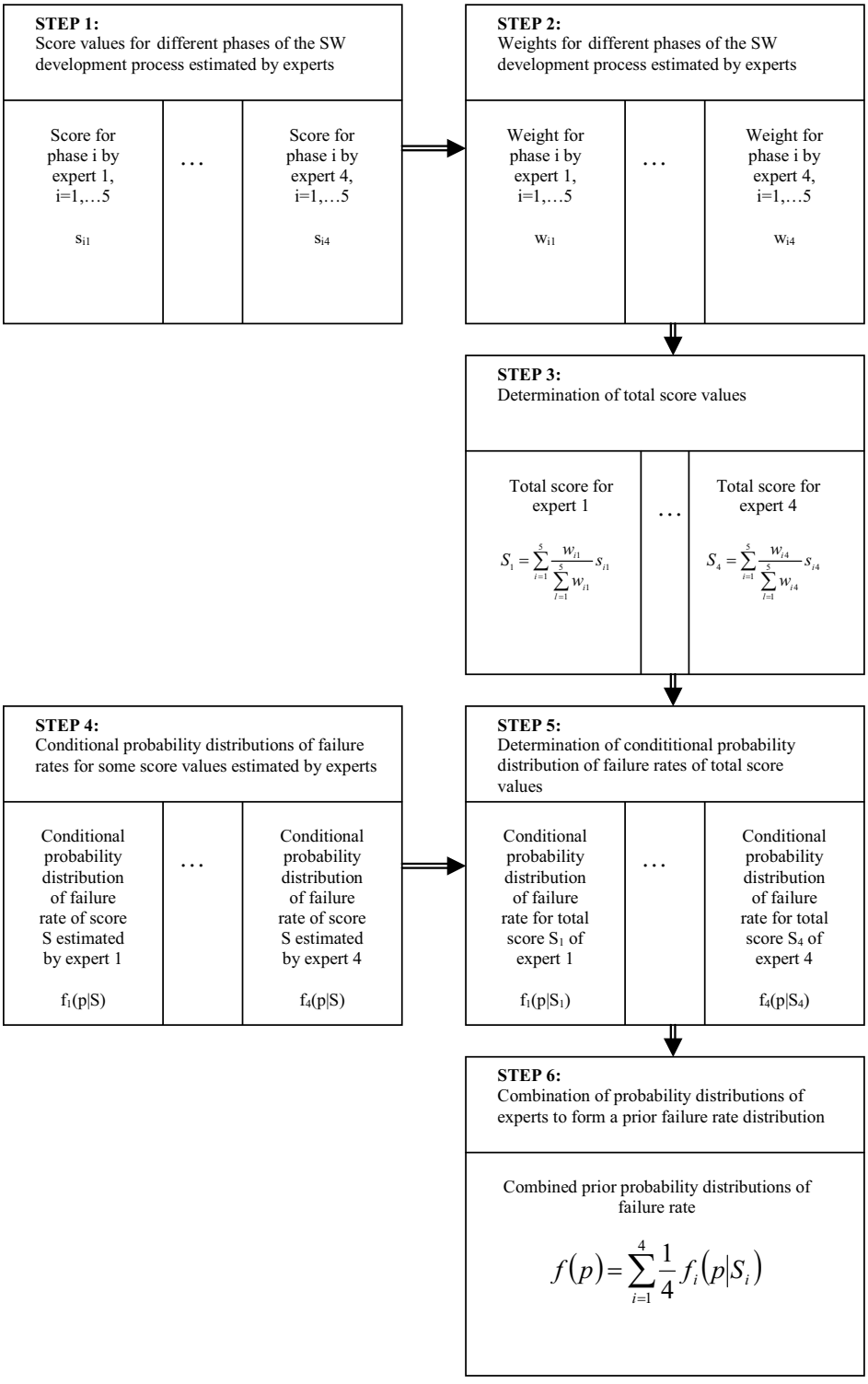

Figure 10. Diagram presenting the six steps of the expert judgement process to estimate the prior failure probability of software.

ries based on the different development phases of a normal computer-based system. The categories were: project control and quality, requirement specification phase, design phase, implementation phase and testing phase. Before the interviews the experts received a short period of training on the assessment process and how to give probability estimations based on expert judgement overall. The training session was given to all experts simultaneously, but the interviews were carried out individually.

In the first two steps of the expert judgement process the different phases of the software development process were discussed. Based on the previous experience and the conclusions of the interviews the experts gave score values and weights for the different software development phases. The score values and weights were numerical values between zero and ten reflecting the expert's opinion on how well the production team managed in the execution of a phase and how big of importance did a certain phase have to the total reliability of the system. In the training session the numerical scale used in the assessment and the meaning of different score values were discussed from the system reliability point of view, and this way each expert was able to build an interpretation of his own about the meaning of the score values. In the third step the values given in the previous two steps were united to a total score value using an additive value function.

In addition to the score values and weights the experts were asked to give failure rate distributions for two or three different score values in step four. The system failure rate distributions should represent expert's opinion of score values as well as possible, since these distributions were used to calibrate the scale of score values. After the calibration it was possible to extrapolate a failure rate distribution for other score values in the scale. The conversion from the total score value to corresponding failure rate distribution was carried out in step five. In the final sixth step of the expert judgement process the reliability estimates or failure rate distributions of all experts were combined together to form the prior failure rate distribution of the first software version of the target system. Detailed presentation of the numerical values derived in the expert judgement process is given in report by Helminen & Pulkkinen [16].

4.3.2 Reliability changes between software versions

In case there are several software versions produced during the life cycle of a computer-based system it is important to be able to use the evidence from previous software versions for the reliability estimation of the last software version. This is especially important if the operational experience for the last software version is small. However, to be consistent in the estimation the reliability changes between successive software versions need to be evaluated.

In the assessment the effects of the software modifications between successive software versions and the criticality of the modifications to the system reliability were evaluated in co-operation between the system developers and the assessment executives. The reliability changes between successive software versions were modelled so that the reliability estimate of a software version was the same as the reliability estimate of the preceding software version added with a normal distributed random change (see Equation 2). A prior estimate on the magnitude of the change was determined depending on the amount and criticality of changes made between software versions.

4.3.3 Operational experience

The second main evidence source in the assessment was the operational experience estimated during the life cycle of the software-based system. The operational experience was the approximated amount of operating years and the amount and types of software defects encountered for the different software versions of the system. The software defects were classified to software faults and software inconveniences depending on the severity of a defect from the customer's point of view. The faults and inconveniences were reported either by the developer or the customers. After detection, a defect was analysed and depending on the nature of the defect it was corrected in the next software version.

As well known, the reliability of a computer-based system is a factor of two properties. First of all, it is a property of the faults the system may contain. However, even though there may be faults in the software these faults are only revealed when certain inputs are introduced to the system.

The probability distribution of input sequences introduced to the system varies from one operational profile to another. The reliability of a computer-based system is, therefore, a property of the system faults and the operation profile the system is functioning in. To simplify the modelling and the calculations in the assessment it was assumed that the estimated operational experience for different software versions was obtained from a single and similar operational profile.

4.4 Sensitivity analysis and results

In the assessment sensitivity analyses concerning two different subjects were carried out. In the first sensitivity analysis the influence of the prior reliability changes between software versions was evaluated. In the second analysis the significance of the interpretation of the software defects to the results was determined. Due to the different sensitivity analyses there are four different scenarios of the final results.

In the sensitivity analysis of reliability changes two different approaches on the influence of software changes were taken. Calculations were carried out using a neutral and a conservative approach. In the neutral approach the prior mean value of the change term between successive software versions was assumed zero. In the conservative approach it was assumed as a prior assumption that a change in software has a negative influence to the reliability of the system. What this means is that as a fault is removed from the software new faults are always introduced to the software. To evaluate the influence of different interpretations of software defects to the results two different data sets were used. In the first data set only the number of software faults encountered for different software versions were implemented to the model as fault data. In the second data set both the software faults and the software inconveniences were taken into count.

In the life cycle of the motor protection relay there had been seven different software versions. The software versions are labelled from A to G as shown in Figure 9. Calculated posterior failure rate distributions for different software versions using the conservative approach are presented in Figure 11. The posterior failure rate distributions range from 2.5 percentile, the lower bar, to 97.5 percentile, the upper bar, and median marked as a dot somewhere in between. Corresponding graphs for the neutral approach are shown in Figure 12.

From the graphs it can be seen how the confidence on the reliability of the motor protection relay is increased while the defects are removed and the system software is updated. For example from Figure 11 it can be approximated that the median value of software version A implies that two times out of thousand, a device will encounter a software fault during a year of operation. For software version G the corresponding median value is four times out of one hundred thousand. In the calculations all the devices were assumed to function in a similar operational profile.

Significant differences between the posterior failure rate distributions of the two approaches used for the influence of software changes cannot be noticed. With the conservative approach the failure rate distributions of different software versions seem to be more monotonous, i.e. the estimates for the early software versions are better than in the neutral approach and vice versa for the later software versions. However, the difference between the two approaches for the failure rate of the last and crucial software version is negligible as can be verified from the figures. Explanation to the small difference in the last software version can most probably be found from the large amount of operational experience data, and thereby the dominant role of the operational experience in the assessment.

In general, the posterior failure rate distributions of the software versions provide an informative way to follow up the evolution of reliability trend during the life cycle of the computer-based system. From the figures of the assessment it is easy to estimate when the system has reached a maturity required by a certain application.

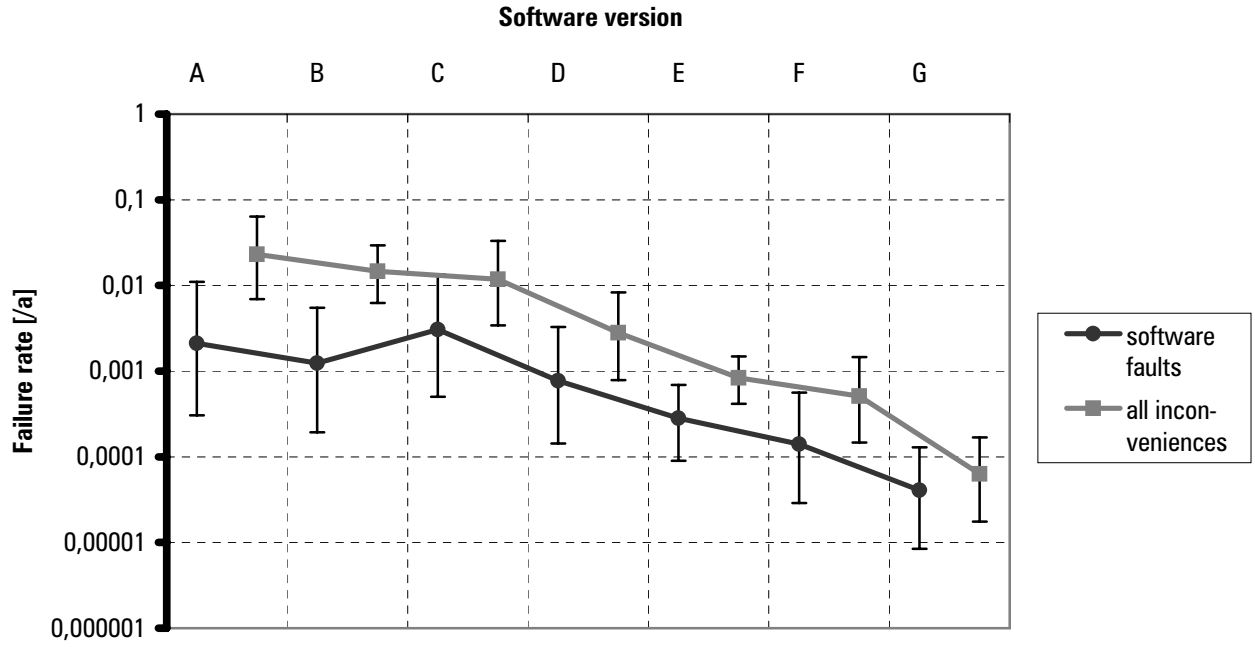

Figure 11. Posterior 2,5–50–97,5 percentiles for failure rate distributions of different software versions of the conservative approach.

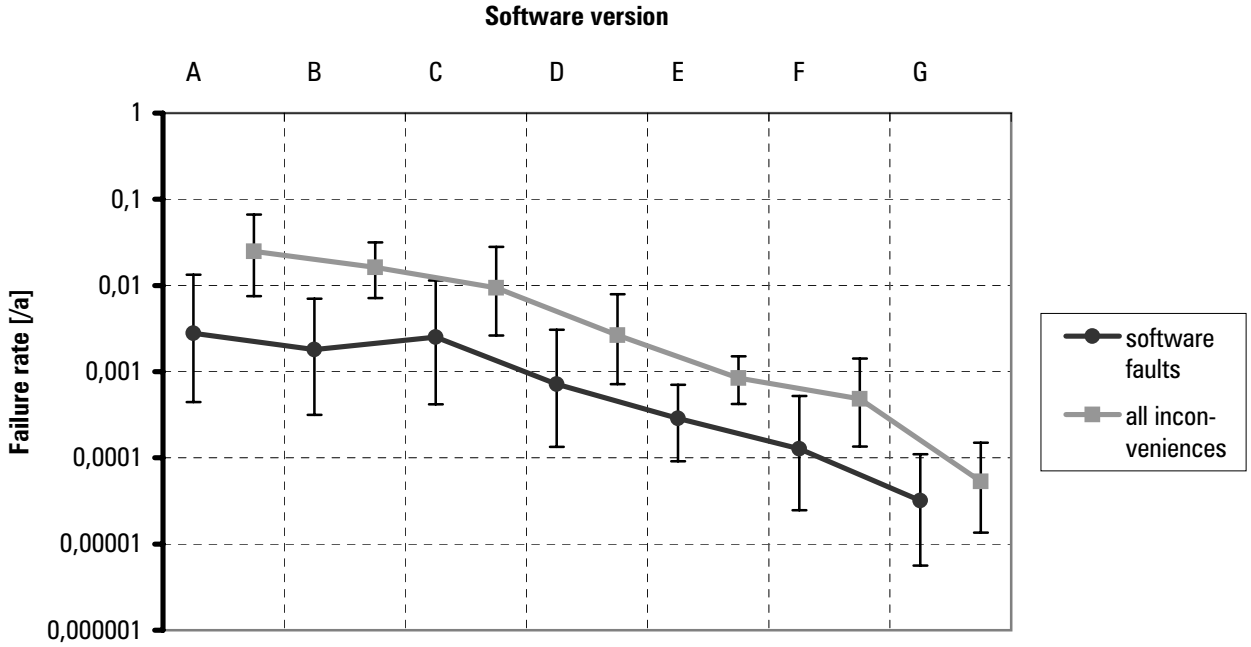

Figure 12. Posterior 2,5–50–97,5 percentiles for failure rate distributions of different software versions of the neutral approach.

5 Discussion

5.1 Quantitative reliability assessment in safety case

As already stated above a safety related system must have a safety case, which provides a body of evidence that the system is adequately safe for a given application and environment over its lifetime. The safety case should include an assessment of the functionality, performance and reliability of the system. The safety demonstration of computer-based systems is a challenging task and methods for this purpose have been developed worldwide. However, approaches for quantitative reliability analysis of computer-based systems are still in a phase of early development. PASSI-project approached the reliability analysis by applying Bayesian networks.

Conventional quantitative reliability assessment aims at determining an estimate for a failure probability of a system. In principle, the reliability of computer-based systems can be seen analogous to that of mechanical systems. It is fully meaningful to ask for a probability that a computer-based system does not operate as specified in certain situations. This becomes evident for example in probabilistic safety assessment (PSA), where the failure probabilities for safety functions are needed independently on the technology by which the functions are realised.

The probabilistic interpretation of reliability, i.e. the uncertainty concerning the occurrence of failures expressed in probabilistic concepts, is a rational approach in reliability and risk assessments. For example in PSA the failure probability of a safety function is estimated by developing a fault tree model, determining the probabilities of the basic events (i.e. the component failures), and calculating the final failure probability estimate using the rules of probability calculus.

However, the construction of the fault tree for a computer-based safety function may be a problematic task due to many reasons. First, a computerised system is often designed to perform several safety functions. PSA requires failure probabilities for each of these functions. In computerised safety systems, these single safety functions use the resources of the platform and they thus are not fully independent. The evidence collected from the design life-cycle of the application and platform refer mainly to the whole of the system, not to the single safety functions. The same is true for operational experience.

Secondly, PSA requires probability that a safety function does not operate correctly when demanded. In principle, the demand situations are defined in the requirements specification of the system. However, the concept of demand is problematic here: it is not always clear whether the evidence (operating experience or evidence from design features and design process) really correspond exactly the demand situation. Another possibility is that the erroneous operation of a safety function causes an initiating event. In that case one has to identify such erroneous operation modes, which can cause an initiating event. The requirement specifications are connected to this by defining the fail-safe states of the system. In PSA, these failure events are quantified by determining the failure rate or intensity. Again, it is not always clear how the available evidence corresponds to the quantifiable entity.

In the case study described above, an estimate of a failure rate of a single programmable component was determined by using a Bayesian model. A failure of such component is usually a basic event in a fault-tree model of larger safety system or function. The Bayesian network approach yields a reliability estimate, which can be used in a fault tree model for a single component. To develop a reliability estimate for an automation system with several safety function requires, however, much

deeper analysis. The conclusions made on the basis of the case study may not be relevant for the analysis of such systems.

The assessment discussed in this report follows the conceptual approach of SHIP-model providing solid ground for the reliability argumentation of computer-based systems. In a Bayesian model the reliability claims of a system are rationally constructed and the arguments the claims are based on are clearly stated. Assessment method can thereby be used as a tool of gathering a safety case for a system and as a communicative instrument in a licensing process of computer-based automation systems.

5.2 Analysis of evidence by Bayesian networks

One basic idea of Bayesian network model is to describe the evidence and the relationships between various pieces of evidence in the form of probability statements. In principle, each assumption concerning or presenting an evaluation of systems properties should correspond a node (i.e. a random variable) in the Bayesian network model. Thus, in an ideal case, the structure of a Bayesian network should correspond the form of the totality of evidence and assumptions about the system under analysis.

Objective of the case study was on the methodological development of the assessment method, therefore, the evidence on the systems technical features and on the design process were not well structured. Due to this, the evidence on these features were elicited and quantified by expert judgement, and presented as a prior distribution of the failure rate. The structure of the evidence was not described by a network of random variables but as a single probability distribution. This was a feasible modelling decision corresponding conceptually the evidence that was available.

The model of the case study included a description of the operating experience from successive versions of the system under analysis. An important assumption made here was that the operational profiles of each system version were similar. This assumption was thought to be certainly true, in order to simplify the model (although it is fully possible to describe uncertainty of this assumption by suitable random variables). Another assumption was that the failure rate of a version of the system depends in a simple way on that of the preceding version. However, this dependence was assumed stochastic, and its strength could be controlled by certain variables. In the extreme case, the failure rates of successive versions could be made independent, which means that the evidence from earlier versions has no impact on the failure rate of next versions. This means also that the prior evidence (described by the prior distribution) has no impact. In fact this kind of extreme assumption is most conservative: it means that there is no prior evidence, and that the uncertainty is in this sense maximal. The aim of the case study was not to analyse of the truth of these assumptions but to demonstrate the possibilities of Bayesian network methodology to deal with such assumptions.

In a parallel project, which was carried out as a shared cost EU-project (BE-SECBS, see e.g. [18]), some steps of the method were applied to a set of programmed safety functions. The analysis covered an evaluation of the evidence provided by the system vendor and the construction of a Bayesian network model. However, the reliability of the safety functions was not estimated on the project. The Bayesian network followed the structure of the life-cycle of the system under analysis. The application life-cycle steps included into the model were requirement specification, concept design, detailed design, application C-code generation, simulation tests, code compilation and linking and integrated system tests. The variables of the network were some kind of quality ratings of the life-cycle step products and corresponding development processes. Both the construction of the network structure and the quantification of the quality rating were carried out as an expert judgement process. The quantification consisted on developing quality ratings for the process of above mentioned life-cycle steps and the quality evaluation of the results of these steps. The evaluation was made in terms of probability distributions. However, the reliability of the safety functions was not evaluated quantitatively. Thus, the main experience gained from this project was the qualitative analysis of evidence from the development life-cycle and the development of the Bayesian network structure. Although the use of the development life-cycle as the structural basis for the Bayesian network seems a natural approach, there

are still open questions. The evaluation and quantitative measurement of the life-cycle process seems possible, but it is not straightforward to identify the relationship between the measurements and the reliability estimates. The evidence concerning system structure (or the result of an life-cycle step) is more difficult to be included into the Bayesian network model: it is not easy to describe this in the form of random variables, which are needed in Bayesian network. Similarly, the results of analyses (e.g. FMEA, formal analyses of the code correctness, etc) are not easily described as the nodes of Bayesian network.

It seems that the quantitative reliability analysis of programmable systems remains an expert judgement problem. Although operational experience may be available, the relevance of this statistical evidence should be evaluated by an expert team, as noticed in the case study. Similarly, the more or less qualitative evidence about the system itself and from the systems development life-cycle, should be modelled and quantified in an expert judgement process. In this phase of the research, there are no well-established principles to build a Bayesian network model corresponding to this qualitative evidence. The development of such principles requires, at least to some extent, the formal analysis of semantic and logical properties of the evidence, for example along the thoughts discussed by Courtois [19].

In practical cases, Bayesian networks could be constructed to follow the systems development life-cycle models, i.e. the phases of the life-cycle models could be described by suitable random variables. The dependencies between consecutive life-cycle phases could be explicitly described in a model. The analysis of logical structure of the evidence could support also this task. However, this would call for extra understanding of the system from the modeller, which usually hasn't be directly involved in the development of the system. One way to overcome the difficulty is to enhance the communication between the reliability modellers and the experts involved with the development process of the assessed systems.

In the case study a special expert judgement process was carried out for the quantification of the prior reliability estimation of the first software version. The expert judgement process was applied in rather informal way giving the expert opinions lots of influence. Other possibility would have been to use independent assessors. Use of independent assessors would probably have set the documentation of development process and system features in more central role in the prior estimation. However, in the results it was concluded that the significance of the prior estimation had only little influence to the final results because of the dominant role of the operational experience. The results provide a good example of the benefits of using Bayesian networks in the reliability estimation of computer-based systems. In the assessment different pieces of evidence can be compared and their significance to the final results can be estimated.

In spite of the difficulties discussed above, Bayesian networks provide a promising way of finding plausible quantitative reliability estimates of computer-based systems. In an estimate properties and failure behaviours of a system are analysed in an efficient and transparent way. Different pieces of evidence are combined and statistical inference on the evidence is concluded using the consistent framework of Bayesian modelling theory. In the reliability assessment all uncertainties and assumptions are expressed in the form of probability statements and Bayesian interpretation of probability is applied throughout the model. Bayesian network is a model incorporating and combining the various pieces of evidence as a well-defined probability model. Results of the model are presented as informative posterior distributions on parameters of interest.

5.3 Topics for further research

The PASSI-project has covered several research topics in the field of nuclear power plant automation. Such issues as ageing of automation systems, FMEA [20] and reliability analysis of programmable automation systems have been covered. This report deals mainly with the reliability analysis and its role in licensing of the automation system.

The licensing process of automation system is complex and includes several different approaches and analyses. The methodology for these analyses is still developing, and there is lot of needs for further research. Some issues for further research in the area of reliability analysis are discussed here.

The Bayesian network approach applied and

development in the PASSI-project for the licensing of computer-based safety systems is still rather far from being established method. It has been applied only in rather limited cases, which do not correspond the real practical situation. New real life applications are needed. Case studies concerning both single programmed devices and whole programmed safety functions are needed. To support the real life licensing process the studies should concentrate on the most demanding applications on the highest safety class levels. The evidence to be modelled and analysed should be rich enough: data from both operating experience and design features and processes should be covered.

Only the case studies can demonstrate the feasibility and applicability of the approach, and only through them it is possible to develop practical expert judgement techniques needed. However, the reliability analysis is not a pure expert judgement process. In order to reach a practical reliability analysis tool, the experiences and techniques of formal evidence analyses (see [19]) must be combined with the Bayesian network modelling. The logical structure of the safety case should be taken into account in order to analyse the relevance of the evidence. In fact, the Bayesian network structure should be compatible with the formal structure of the safety case. A lot of research work is needed to solve this task.

The licensing process usually includes use of computerised code analysis and requirement analysis tools. The results of this type of analyses form an important and useful body of evidence, which has not been covered in the PASSI-project. This is a weakness, which should be taken into account in future research. Also tools and techniques for requirement management are needed.

The role of quantitative reliability assessment is evident in licensing of computer-based systems. It seems that the expert judgements and subjective probability statements and evaluations will be inevitable. This implies that the interpretation and use of these analyses should be clarified and practises for real risk-informed licensing principles should be demonstrated.

References

1. Decision of the Council of State on the general regulations for the safety of nuclear power plants. 14 Feb. 1991/395.

2. Guide YVL 2.8. Probabilistic safety analysis in safety management of nuclear power plants. STUK, Helsinki, May 2003.

3. Helminen, A. & Pulkkinen, U. Programmable automation system safety assessment (PASSI): PASSI summary report. In FINNUS Final Report, VTT Research Notes 2164, Espoo 2002, pp. 211–217.

4. IEC 61508. Functional safety of electrical/electronic/programmable electronic safety-related systems

5. http://www.adelard.co.uk/.

6. De Gelder, P. Deterministic and probabilistic safety analyses: To which extent are they complementary? AV Nuclear. 1997. 8 pp.

7. Haapanen, P., Korhonen, J. & Pulkkinen, U. Licensing process for safety-critical software-based systems. STUK-YTO-TR 171. Helsinki, 2000. 84 pp.

8. Software for computers in the safety systems of nuclear power stations, Part 2 of IEC 60880.

9. Korhonen, J., Pulkkinen U. & Haapanen P. Diversity requirements for safety critical software-based automation systems. STUK-YTO-TR 142. Helsinki 1998. 48 pp.

10. Bishop, P. & Bloomfield, R. 1995. The Ship Safety Case approach: a combination of system and software methods. ENCRESS / CSR Conference, Safety and Reliability of Software Based Systems. Bruges, Belgium, September 1995.

11. Bloomfied, R.E., Bishop, P., Littlewood, B. & Strigini, L. Safety assessment of hazardous industrial processes in the presence of design faults, Final report, SHIP/D/005. 1995. 30 pp.

12. Korhonen, J., Pulkkinen, U. & Haapanen, P. Statistical reliability assessment of software-based systems. STUK-YTO-TR 119. Helsinki, 1997. 31 pp.

13. Helminen, A. Reliability estimation of safety-critical software-based systems using Bayesian networks. STUK-YTO-TR 178. Helsinki, 2001. 23 pp.

14. Myllymäki, P. & Tirri, H. Bayes-verkkojen mahdollisuudet (in Finnish), National Technology Agency of Finland. Helsinki, 1998. 110 pp.

15. O'Hagan, A. Lecture notes on course: Bayesian statistics – Theory and practice. Helsinki, 2000.

16. Helminen, A. & Pulkkinen, P. Reliability assessment using Bayesian networks – Case study on quantitative reliability estimation of a software-based motor protection relay. STUK-YTO-TR 198, 2003. 31 pp.

17. Musa, J.D., Iannino, A. & Okumoto, K. Software reliability – Measurement, prediction, application. McGraw – Hill Book Company. New York, 1987. 621 pp.

18. V. Kopustinskas, C. Kirchsteiger, B. Soubies, F. Daumas, J. Gassino, JC. Péron, P. Régnier J. Märtz, M. Baleanu, H. Miedl, M. Kersken, U. Pulkkinen, M. Koskela, P. Haapanen, ML. Järvinen, HW. Bock, W. Dreves. *Benchmark Exercise of Safety Evaluation of Computer Based Systems (BE-SECBS)* Shared cost action / FIKS-CT-2000-00054.

19. Courtois, P-J. Semantic structures and logic properties of computer-based system dependbility cases. Nuclear Engineering and Design 203 (2001), pp. 87–106.

20. Haapanen, P. & Helminen, A. Failure mode and effects analysis of software-based automation systems. STUK-YTO-TR 190. Helsinki, 2002. 35 pp.